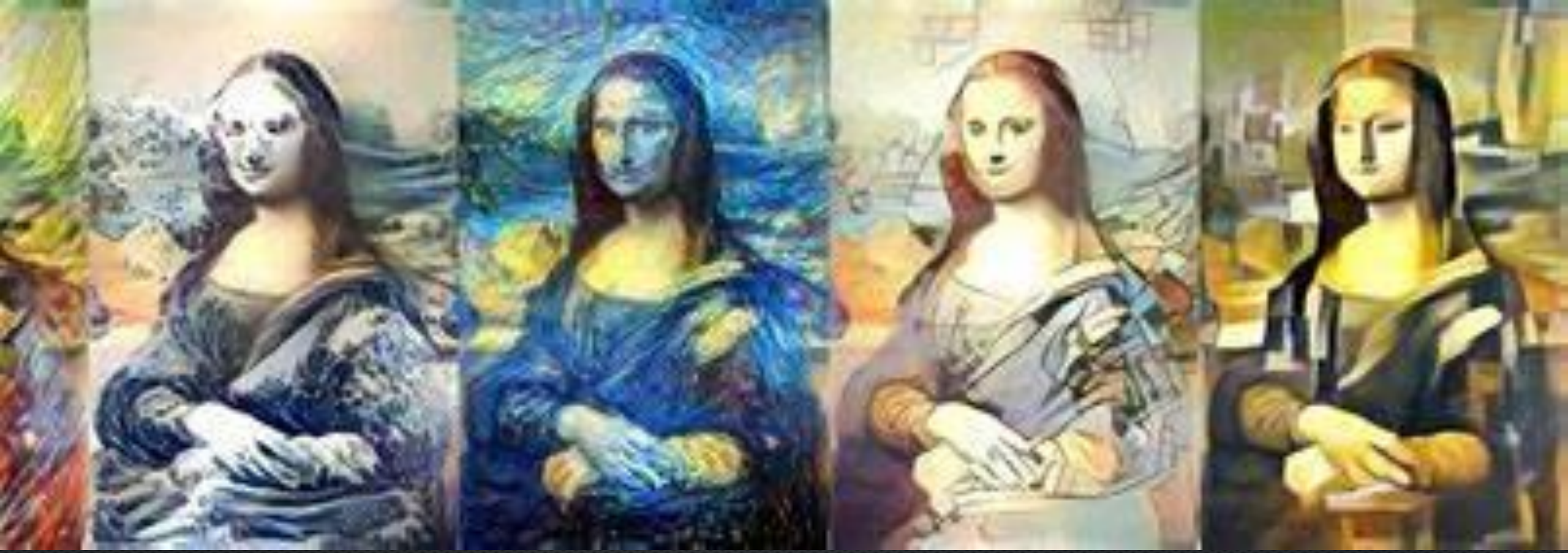


NEURO – STYLE & RECOGNITION

SHAURY SHOBIT

PRAGYA YADAV

TOPICS

- Introduction
- Related work
- Methodology
- Results and Analysis
- Conclusion
- Future Scope

INTRODUCTION

- Programming language Python.
- Neural Network And Datasets project.
- It is combination of more than one idea.
- Neural style transfer + Model generation + Digit recognition + Random facts.
- Light weight.
- Handy and Powerful.

NEURAL STYLE TRANSFER

The intention was to assist my friends who are interested in using style transfer for generating painting styles, providing them with a tool to visualize their desired styles within their scenes.

- **Problem Was :** The issue was that they lacked technical expertise and found it daunting to navigate the programming required to run such a program, despite seeing its potential on Instagram.
- **Solution :** The solution involved creating a simplified version of the program tailored to their needs, with a user-friendly interface that requires minimal technical knowledge, enabling anyone to utilize it effectively.

DIGIT RECOGNITION

THE AIM WAS TO FACILITATE CHILDREN'S LEARNING THROUGH ENJOYABLE ACTIVITIES, EMPOWERING PARENTS TO TEACH THEM HOW TO WRITE DIGITS INDEPENDENTLY WHILE ALSO INTRODUCING THEM TO ESSENTIAL RANDOM FACTS FOR THEIR KNOWLEDGE DEVELOPMENT.

- **PROBLEM WAS :** THE CHALLENGE LAY IN CREATING A MODEL AND SEAMLESSLY INTEGRATING IT INTO A RECOGNITION PROCESS WITH A USER-FRIENDLY INTERFACE.
- **SOLUTION :**
  - NEURAL NETWORK TO MAKE A MODEL.
  - USING EXISTING DATASET BY TENSORFLOW MNIST WHICH IS A PREBUILT DATASET.
  - GRADIO FOR MINIMALISTIC UI WHICH IS BEST SUITABLE FOR INTEGRATION OF ML AND UI

RELATED WORK

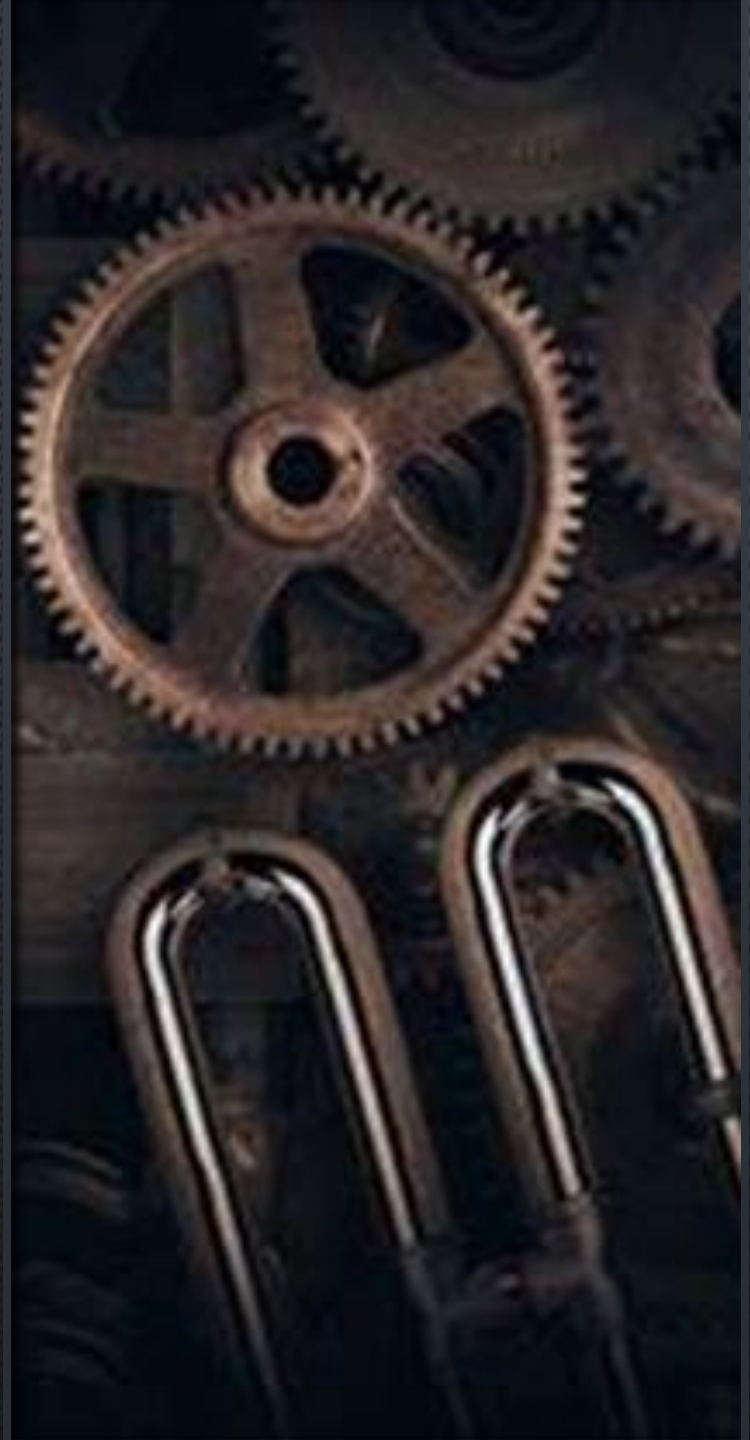
- "A Neural Algorithm of Artistic Style" by Leon Gatys et al., originally released to ArXiv 2015
- "Understanding VGG19" - by various websites (researchgate.net ; arxiv.org ; iq.opengenus.org)
- Github - anishathalye/neural-style

METHODOLOGY

- COLLECTED RELATED WORKS.
- LEARN & UNDERSTAND THE PROJECT MODULES & LIBRARIES.
- SELECTION OF CORRECT LIBRARIES.
- STARTED BY BREAKING DOWN THE PROJECT.

CONTD.

- 3 MODULES:
  - NEURAL STYLE TRANSFER
  - DIGIT RECOGNITION
  - UI – TKINTER & GRADIO
- TEST & ANALYZE EACH MODULES.
- FIXED THE ERRORS.



WORKING

- Input Image: Image provided by the user.
- Processed Image: Intermediate image after style transfer.
- Recognized Digit Data: Result of digit recognition.
- Stylized Image / Recognized Digit: Final output to the user.

WORKING

Style Transfer

- Existing Work
- Compatibility of Hardware, Software & System.
- CNN selection - VGG19
- New Process - implementation
- UI integration
- Testing

Digit Recognition

- MNIST dataset
- Compatibility of Hardware, Software & System.
- Model Generation
- UI integration
- Testing

VGG-19

- Developed by the Visual Graphics Group at Oxford
- VGG19 is a CNN architecture that has 19 layers, including 16 convolutional layers and 3 fully connected layers.
- VGG19 is particularly valuable because it can capture intricate details of both content and style from images.
- By leveraging pre-trained VGG19, we can efficiently apply complex styles to content images

STYLIZATION

- Content and Style Images selection
- Use a pre-trained convolutional neural network VGG19, to extract features from both the content and style images.
- Feature Extraction - Content and Style
- Optimization - Content and Style Loss, Total variational loss
- Backpropagation and Update
- Final Stylized Image

MNIST

- The MNIST dataset consists of 70,000 grayscale images of handwritten digits, ranging from 0 to 9.
- Image Size: Each image in the MNIST dataset is 28x28 pixels.
- Minimal preprocessing is required for the MNIST dataset as it is already normalized and centered.
- The simplicity and reliability of the dataset make it a go-to for educational purposes and benchmarking

CNN

- CNNs) play a crucial role due to their powerful feature extraction capabilities.
- Extract high-level features from a designated layer
- Multiple layers of the CNN to capture different aspects of the style, such as textures and patterns.
- The generated image is iteratively updated by minimizing a combination of content loss and style loss
- Use of pooling to reduce spatial dimensions.

MODEL GENERATION

- The model is a CNN, chosen for its ability to effectively capture spatial hierarchies in image data.
- The model is trained and tested on the MNIST dataset, which consists of 60,000 training images and 10,000 test images of handwritten digits (0-9). Each image is 28x28 pixels in size.
- Uses ReLU (Rectified Linear Unit) to introduce non-linearity, which helps the network learn complex patterns.
- Dense Layer is to processes the flattened features, combining them to form the final output.

RESULTS & ANALYSIS

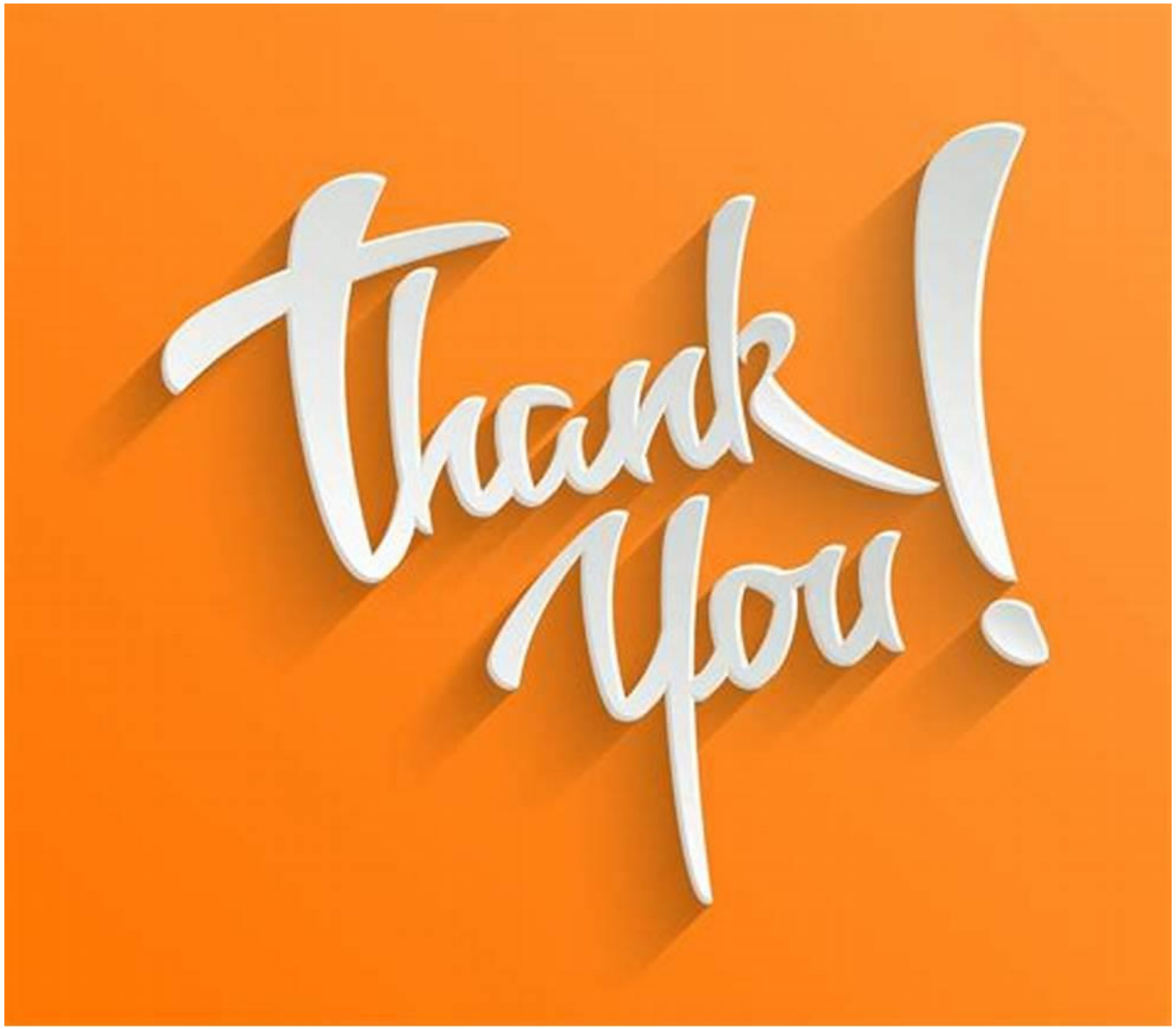
- In each module we faced different issues.
- Analyzing each part of code to make it simpler to understand and run efficiently.
- Tested it at every stage.
- Worked on our customer feedbacks.

Conclusion

- Covered our course.
- This project was a fun and learn process.
- Learn some new and useful stuff.
- Working of Neural Networks and Tensorflow.

FUTURE SCOPE

- Improve UI.
- Add more functionalities. - Widen the Learning scope(English, Hindi characters, colors, objects, and much more)
- Improve algorithm calculations.
- Style Transfer for high end images
- Model for various
- More accurate and Faster response.


Thank You!